

PENGATURAN POSISI MOTOR SERVO DC DENGAN METODE P, PI, DAN PID

Nanang Budi Hartono, Kemalasari, Bambang Sumantri, Ardik Wijayanto

Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya

Kampus PENS-ITS Sukolilo, Surabaya

luffy@student.eepis-its.edu

lecturer.eepis-its.edu/~kemala

lecturer.eepis-its.edu/~bambang

lecturer.eepis-its.edu/~ardik

Abstrak— Motor DC merupakan salah satu jenis aktuator yang cukup banyak digunakan dalam bidang industri atau robotika. Pada mobile robot line tracer misalnya menggunakan motor DC sebagai aktuatornya.

Pada proyek akhir ini akan dibuat motor servo DC sendiri dengan menggunakan motor DC yang telah dipasang mekanik sedemikian rupa dengan gear pembanding agar putarannya tidak berputar dengan cepat dan dapat diatur. Dan pada pemasangan motor DC juga dipasang potensiometer sebagai sensornya.

Untuk komunikasi antara PC dengan modul proyek akhir ini digunakan PCI 1712 yang dimanfaatkan adalah bagian analog input, analog output dan digital output. Untuk menggerakkan motornya dibuat rangkaian PWM analog dan rangkaian pendukung lainnya dalam modul ini. Output respon sistem pada proyek akhir ini antara lain membandingkan penggunaan plant saat menggunakan kontroler P, PI, PID dan menggunakan PID yang sudah diberi pembebanan. Rata-rata error pembacaan pada busur derajat yang didapat didapat pada proyek akhir ini sebesar $0,5833^{\circ}$.

Kata kunci : motor DC, P, PI, PID, rangkaian PWM analog, PCI 1712

I. PENDAHULUAN

1.1 Latar Belakang

Motor DC merupakan salah satu jenis aktuator yang cukup banyak digunakan dalam bidang industri atau robot. Pada mobile robot line tracer misalnya menggunakan motor DC sebagai aktuatornya. Selain harganya murah juga cukup handal penggunaannya. Pada bidang industri juga banyak seperti pada conveyor menggunakan motor DC juga.

Dalam TA ini akan dibahas bagaimana cara mengatur posisi motor servo DC dengan menggunakan kontroler P, PI, dan PID. Mungkin dalam Penggunaan metode PID dalam kontrol motor sudah banyak sekali digunakan. Terutama kontrol motor DC yang banyak sekali aplikasinya di dunia industri. Biasanya dalam kontrol kecepatan motor. Dalam TA ini akan dibuat aktuator motor DC servo sendiri. Motor DC

akan dipasang mekanik sedemikian rupa dengan gear pembanding agar putarannya tidak berputar dengan cepat dan dapat diatur. Dan pada pemasangan motor DC juga dipasang potensiometer sebagai sensornya. Motor DC yang digunakan adalah motor DC dengan tegangan 12 volt. Dengan menggunakan metode kontrol P, PI, dan PID akan dibandingkan masing-masing responsnya dan dianalisa mengapa terjadi demikian.

1.2 Tujuan

Proyek akhir dengan judul “Sistem Pengaturan Posisi Motor Servo Dc Dengan kontroler P, PI dan PID” ini bertujuan untuk membuat suatu modul pembelajaran pengaturan posisi motor servo dc. Dengan adanya proyek akhir ini diharapkan dapat membantu memudahkan mahasiswa dalam memahami Kontroler P, PI dan PID pada praktikum kontrol cerdas.

1.3 Permasalahan

Adapun permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana cara merancang rangkaian PWM analog dengan benar.
2. Bagaimana cara mengkomunikasikan antara PC dengan modul yang dibuat.
3. Bagaimana merancang mekanik motor dc yang dijadikan motor servo dc sendiri.
4. Bagaimana cara penggunaan *visual basic* dalam pengaturan posisi motor.

1.4 Batasan Masalah

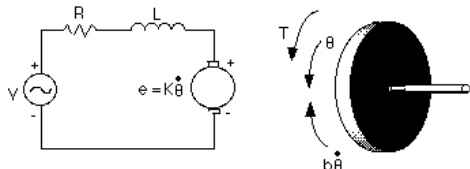
Adapun batasan-batasan masalah yang dibuat agar dalam pengerjaan proyek akhir ini dapat berjalan dengan baik adalah sebagai berikut :

1. Motor servo yang akan dibuat dalam proyek akhir ini adalah *motor servo standart*.
2. Sudut masukan yang diberikan berada pada kuadran 1 dan 2 ($0-180^{\circ}$).
3. Sudut masukan terkecil dari sistem adalah 1° .
4. Menggunakan potensiometer sebagai umpan balik dari sistem.

5. Menggunakan *PCI 1712* sebagai konversi data dan komunikasi antara PC dengan motor.
6. Menggunakan software *visual basic* sebagai masukan dan tampilan grafik respon waktu.

II. TINJAUAN PUSTAKA

2.1 Permodelan posisi motor dc



Gambar 2.1 permodelan rangkaian listrik dari motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan[1] berikut ini :

$$T = K_a i \dots\dots\dots (2.1)$$

dimana K_a adalah konstanta jangkar motor yang bergantung pada banyaknya lilitan pada jangkar, jumlah kutub medan, tipe belitan dan penampang jangkarnya. Adapun besarnya tegangan ggl induksi lawan yang dibangkitkan motor ketika berputar adalah sebanding

dengan konstanta motor K_b dan kecepatan sudut putaran motor $\dot{\theta}$ atau turunan pertama dari posisi sudut motor (θ) :

$$e = K_b \dot{\theta} \dots\dots\dots (2.2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redaman dari motor adalah:

$$T = J \ddot{\theta} + b \dot{\theta} \dots\dots\dots (2.3)$$

Dari persamaan (2.1) dan (2.3) diperoleh

$$J \ddot{\theta} + b \dot{\theta} = K_a i \dots\dots\dots (2.4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah

$$V = Ri + L \frac{di}{dt} + e \dots\dots\dots (2.5)$$

$$V = Ri + L \frac{di}{dt} + K_b \dot{\theta} \dots\dots\dots (2.6)$$

Dengan me-laplace-kan persamaan (2.4) dan (2.6) kemudian mensubstitusikannya, maka diperoleh fungsi transfer antara posisi sudut motor θ terhadap tegangan armature V dimana $K_a = K_b$

$$s(Js + b) \theta(s) = K i(s) \dots\dots\dots (2.7)$$

$$(Ls + R) i(s) = V - K_s \dot{\theta}(s) \dots\dots\dots (2.8)$$

$$\frac{\theta}{V} = \frac{K}{s((Js+b)(Ls+R) + K^2)} \dots\dots\dots (2.9)$$

Dimana :

R = Resistansi jangkar

L = Induktansi jangkar

i = Arus jangkar

e = *back emf* (tegangan yang dihasilkan ketika jangkar berputar dalam medan magnet arus searah)

T = Torsi motor

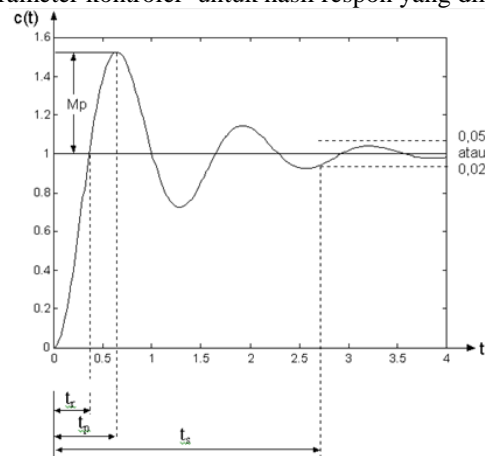
Θ = Perpindahan sudut dari poros rotor

J = Momen Inersia motor

b= Koefisien gesekan viskos motor

2.2 Identifikasi Plant

Identifikasi *plant* ditujukan untuk mendapatkan model matematis berupa fungsi alih yang digunakan untuk proses perancangan kontroler nantinya. Untuk jenis *plant* yang dibahas berupa motor DC sistem orde dua. Orde sistem menentukan jenis kontroler yang akan dipakai dan mencari nilai parameter kontroler untuk hasil respon yang diinginkan.



Gambar 2.2 Respon Orde Dua Terhadap Masukan *Unit Step*

Untuk persamaan respon orde dua menggunakan persamaan sebagai berikut:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \dots\dots\dots (2.10)$$

Sedangkan untuk mencari nilai ζ dan ω_n menggunakan persamaan sebagai berikut :

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \dots\dots\dots (2.11)$$

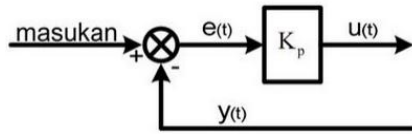
$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \dots\dots\dots (2.12)$$

2.3 Kontroler P, PI, dan PID

Kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal *error*), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara *input* kontroler $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 2.13.

$$u(t) = K_p e(t) \dots\dots\dots(2.13)$$

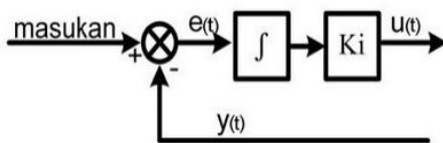
K_p adalah konstanta proporsional. Diagram blok kontrol proporsional ditunjukkan pada gambar dibawah ini



Gambar 2.3 kontrol proporsional

Kontrol integral pada prinsipnya bertujuan untuk menghilangkan kesalahan keadaan tunak (*offset*) yang biasanya dihasilkan oleh kontrol proporsional. Hubungan antara *output* kontrol integral $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 2.14.

$$u(t) = K_i \int_0^t e(t)dt \dots\dots\dots(2.14)$$

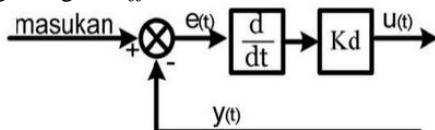


Gambar 2.4 kontrol integral

Kontrol derivatif dapat disebut pengendali laju, karena *output* kontroler sebanding dengan laju perubahan sinyal *error*. Hubungan antara *output* kontrol derivatif $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 2.15.

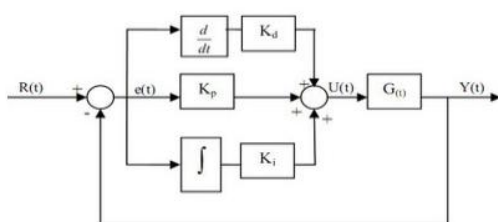
$$u(t) = K_d \frac{de(t)}{dt} \dots\dots\dots(2.15)$$

Blok kontrol derivatif ditunjukkan pada Gambar 2.5. Kontrol derivatif tidak akan pernah digunakan sendirian, karena kontroler ini hanya akan aktif pada periode peralihan. Pada periode peralihan, kontrol derivatif menyebabkan adanya redaman pada sistem sehingga lebih memperkecil lonjakan. Seperti pada kontrol proporsional, kontrol derivatif juga tidak dapat menghilangkan *offset*.



Gambar 2.5 kontrol derivatif

Gabungan dari ketiga kontroler tersebut menjadi kontrol PID[2]. Diagram Blok dari kontrol PID ditunjukan pada gambar 2.6.



Gambar 2.6 kontrol PID

Sehingga persamaan untuk kontrol PID adalah

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \dots\dots\dots(2.16)$$

Dengan:

$u(t)$ = sinyal *output* pengendali PID

K_p = konstanta proporsional

K_i = konstanta integral

K_d = konstanta derivatif

$e(t)$ = sinyal *error*

2.4 KONTROLER DIGITAL

Pada persamaan 2.13 sampai 2.16 merupakan persamaan- persamaan dalam domain waktu continuous (analog). Sedangkan agar persamaan persamaan tersebut dapat direalisasikan dalam bentuk pemrograman pada komputer atau mikrokontroler, maka persamaan dalam domain waktu continuous tersebut harus didiskretisasi terlebih dahulu (digital)[2].

- Kontroler Proporsional

Output dari kontroler proporsional adalah perkalian antara *error* dan konstanta proporsional (K_p). Dari persamaan 2.13 jika didiskritkan menjadi

$$u(k) = K_p e(k) \dots\dots\dots(2.17)$$

- Kontroler Integral

Output kontroler integral selain ditentukan oleh konstanta integral juga ditentukan oleh kondisi seluruh *error* sebelumnya. Dari persamaan 2.14 jika didiskritkan menjadi

$$u(k) = K_i T \sum_{i=0}^k e(i) \dots\dots\dots(2.18)$$

Dimana : $u(k)$ = *output* kontroler ke-k

T = *Time sampling*

Σ = jumlah total *error* mulai $i=0$ sampai dengan $i=k$

- Kontroler Derivatif

Output kontroler derivatif ditentukan oleh koefisien derivatif dan selisih antara *error* sekarang dengan *error* sebelumnya.

$$u(k) = K_d \frac{e(k) - e(k-1)}{T} \dots\dots\dots(2.19)$$

Dimana : $u(k)$ = *output* kontroler ke-k

T = *time sampling*

$E(k)$ = *error* ke-k

$E(k-1)$ = *error* saat k-1(sebelumnya)

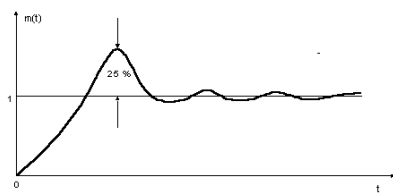
2.5 Menentukan parameter kontroler p, pi dan pid

Penalaan parameter controller PID selalu didasari atas tinjauan terhadap karakteristik yang diatur (*Plant*). Untuk menentukan parameter kontroler P, PI dan PID digunakan suatu metode yaitu metode Ziegler Nichols.

2.6 Metode Ziegler-Nichols

Ziegler-Nichols pertama kali memperkenalkan metodenya pada tahun 1942. Metode ini memiliki dua cara, metode osilasi dan kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan

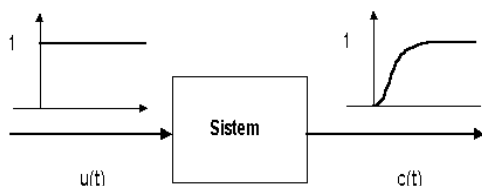
maksimum sebesar 25%. Gambar 2.7 dibawah ini menunjukkan kurva dengan lonjakan 25%. [3].



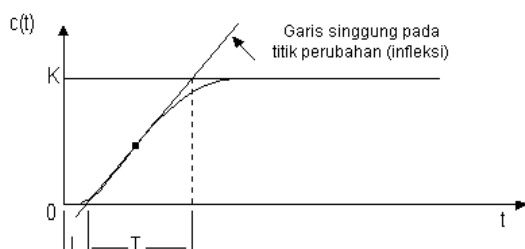
Gambar 2.7 Kurva respons tangga satuan yang memperlihatkan 25 % lonjakan maksimum

2.7 Metode Kurva Reaksi

Metode ini didasarkan terhadap reaksi sistem untai terbuka. *Plant* sebagai untai terbuka dikenai sinyal fungsi tangga satuan (gambar 2.8). Kalau plant minimal tidak mengandung unsur integrator ataupun pole-pole kompleks, reaksi sistem akan berbentuk S. Gambar 2.9 menunjukkan kurva berbentuk S tersebut. Kelemahan metode ini terletak pada ketidakmampuannya untuk *plant* integrator maupun plant yang memiliki pole kompleks



Gambar 2.8 Respon tangga satuan sistem



Gambar 2.9 Kurva Respons berbentuk S.

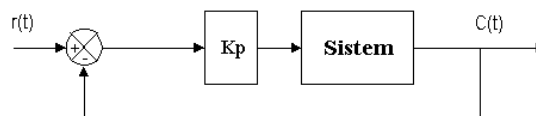
Kurva berbentuk-s mempunyai dua konstanta, waktu mati (*dead time*) L dan waktu tunda T . Dari gambar 2.9 terlihat bahwa kurva reaksi berubah naik, setelah selang waktu L . Sedangkan waktu tunda menggambarkan perubahan kurva setelah mencapai 66% dari keadaan mantapnya. Pada kurva dibuat suatu garis yang bersinggungan dengan garis kurva. Garis singgung itu akan memotong dengan sumbu absis dan garis maksimum. Perpotongan garis singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L .

Tabel 2.1 Penalaan paramater PID dengan metode kurva reaksi

Tipe kontroler	K_p	T_i	T_d
P	T/L	\sim	0
PI	$0,9 T/L$	$L/0,3$	0
PID	$1,2 T/L$	$2L$	$0.5L$

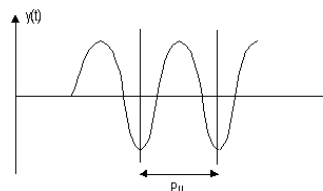
2.8 Metode Osilasi

Metode ini didasarkan pada reaksi sistem untai tertutup. *Plant* disusun serial dengan kontroler PID. Semula parameter parameter integrator disetel tak berhingga dan parameter diferensial disetel nol ($T_i = \infty$; $T_d = 0$). Parameter proporsional kemudian dinaikkan bertahap. Mulai dari nol sampai mencapai harga yang mengakibatkan reaksi sistem berosilasi. Reaksi sistem harus berosilasi dengan magnitud tetap (*Sustain oscillation*) [4]. Gambar 2.10 menunjukkan rangkaian untai tertutup pada cara osilasi



Gambar 2.10 Sistem untai tertutup dengan alat kontrol proporsional

Nilai penguatan proportional pada saat sistem mencapai kondisi *sustain oscillation* disebut *ultimate gain* K_u . Periode dari *sustained oscillation* disebut *ultimate period* T_u . Gambar 2.11 menggambarkan kurva reaksi untai tertutup ketika berosilasi.



Gambar 2.11 Kurva respon sustain oscillation

Penalaan parameter PID didasarkan terhadap kedua konstanta hasil eksperimen, K_u dan P_u . Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i , dan T_d berdasarkan rumus yang diperlihatkan pada Tabel 2.2.

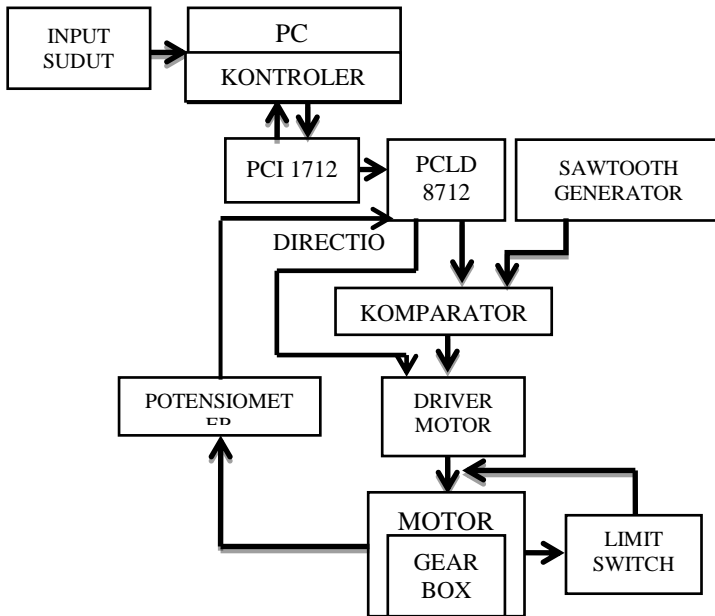
Tabel 2.2 Penalaan paramater PID dengan metode osilasi

Tipe Kontroler	K_p	T_i	T_d
P	$0,5 K_u$		
PI	$0,45 K_u$	$\frac{1}{2} P_u$	
PID	$0,6 K_u$	$0,5 P_u$	$0,125 P_u$

III. PERENCANAAN DAN PEMBUATAN SISTEM

3.1 Gambaran Sistem

Secara umum sistem yang akan dibangun pada proyek akhir ini dapat dilihat pada blok diagram berikut ini:

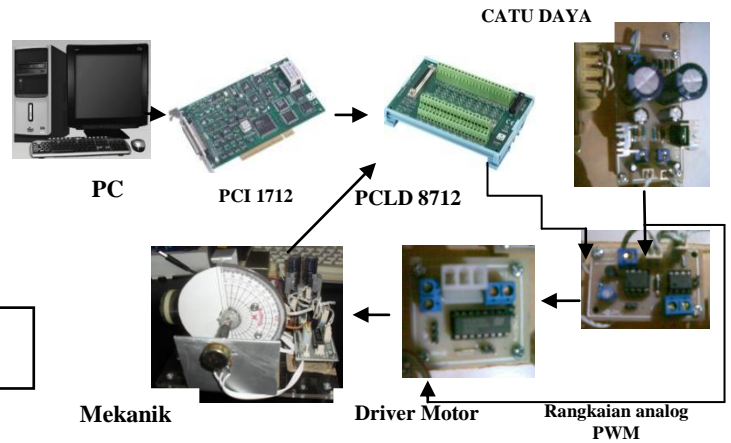


Gambar 3.1 blok diagram sistem

Informasi sudut masukan diolah oleh PC, kemudian *output* dikeluarkan melalui PCI 1712 yang dihubungkan dengan PCLD 8710 sebagai *wiring terminal*-nya berupa tegangan analog yang akan dibandingkan dengan pembangkit gelombang gigi gergaji. Sehingga akan dihasilkan gelombang dengan lebar pulsa yang dapat diatur atau *pulse width modulation*(PWM). PWM ini digunakan untuk mengatur cepat lambatnya putaran motor DC pada driver motor. Selain mengirimkan tegangan analog, PCI juga mengirimkan tegangan digital output sebagai penentu arah putaran motor. Motor DC diaplikasikan dengan gear pembanding dan telah dipasang potensiometer sebagai sensor posisinya dan acrilik berbentuk bulat yang sudah dipasang busur derajat sebagai informasi sudutnya. Saat motor berputar maka potensiometer juga akan berputar. Perubahan nilai tegangan dari perubahan posisi potensiometer ini masuk ke PC melalui PCI 1712 untuk di kontrol kembali yang menjadi informasi sudutnya. Pada proyek akhir ini modul yang dibuat mempunyai range antara 0^0 sampai dengan 180^0 Karena motor DC didesain hanya mampu berputar 180^0 maka dipasang *limit switch* untuk menghentikan putaran motor. Sehingga saat tidak mengenai *limit switch* maka motor akan tetap berputar sedang saat mengenai *limitswitch* maka motor akan berhenti.

3.2 Perancangan Perangkat Keras

Dengan menggunakan komponen-komponen di atas, maka rancangan sistem pada proyek akhir dapat dilihat sebagai berikut:



Gambar 3.2 Komponen Sistem Perangkat Keras

Dari perancangan diatas PC akan dipasang PCI 1712 yang kemudian disambung dengan PCLD 8712 sebagai *card I/O*.. *Input* dari I/O akan masuk rangkaian PWM analog yang terdiri dari 2 macam rangkaian yaitu rangkaian pembangkit sinyal gigi gergaji dan rangkaian komparator. Rangkaian ini akan menghasilkan sinyal PWM sesuai hasil yang diinginkan. Kemudian akan menuju motor driver yang selanjutnya akan masuk mekanik motor yang telah dipasang sensor posisi yaitu potensiometer. Potensiometer ini akan mengirimkan umpan balik kembali yang kemudian akan menjadi informasi ke PC. Dan juga terdapat *limit switch* sebagai pembatas putaran motor.

3.3 Perancangan Perangkat Lunak

Perangkat lunak (software) yang dibutuhkan untuk mengintegrasikan menjadi satu sistem utuh dibangun dengan menggunakan visual basic 6.

Pembuatan perangkat lunak meliputi:

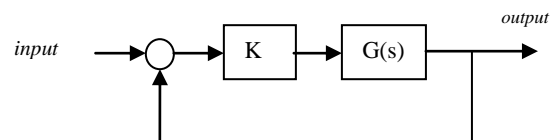
- ❖ Program yang bersangkutan dengan PCI 1712 yaitu analog input, analog output dan digital output.
- ❖ Program kontroler PID
- ❖ Program tampilan respon dari sistem

3.3.1 Program PCI

Sebenarnya PCI 1712 sendiri sudah menyertakan contoh untuk program analog input, analog output dan digital output.

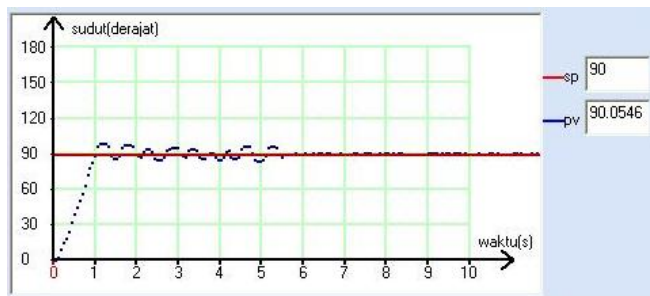
3.3.2 Identifikasi *plant* orde 2

Untuk mendapatkan identifikasi *plant* orde 2 dilakukan dengan menggunakan cara sebagai berikut:



Gambar 3.3 blok diagram close loop

Pada blok diagram diatas kita set nilai K=1 sedangkan G(s) merupakan *plant* yang telah dibuat. Nantinya akan didapatkan respon sistem dari blok tersebut.



Gambar 3.4 respon sistem diagram close loop dengan k=1

Dari gambar 3.4 diatas *setpoint* yang digunakan adalah 90^0 . Dari gambar tersebut didapat %OS nya sebesar 12^0 dan $T_p = 1.2$ detik. Dari data tersebut dapat kita masukkan persamaan untuk mencari G(s).

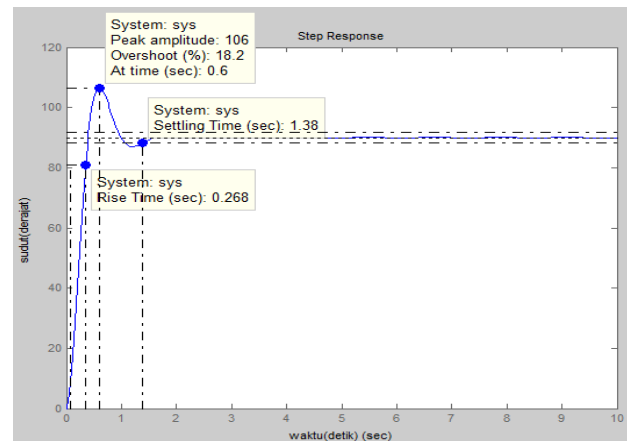
$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Sedangkan untuk mencari nilai ζ dan ω_n menggunakan persamaan

$$\begin{aligned}\zeta &= \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \\ &= \frac{-\ln(3,33/100)}{\sqrt{\pi^2 + \ln^2(3,33/100)}} \\ &= \frac{3,40112}{4,63} \\ &= 0,734599 \\ T_p &= \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \\ \omega_n &= \frac{\pi}{T_p \sqrt{1-\zeta^2}} \\ &= \frac{\pi}{1,2 \sqrt{1-0,734599^2}} \\ &= \frac{\pi}{1,2 \sqrt{0,4604}} \\ &= 4,209\end{aligned}$$

$$\begin{aligned}\text{Jadi } G(s) &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ &= \frac{4,209^2}{s^2 + 2 \cdot 0,734599 \cdot 4,209 s + 4,209^2} \\ &= \frac{17,7165}{s^2 + 5,6687 s + 17,7165}\end{aligned}$$

Kemudian persamaan diatas didapat kemudian dibandingkan hasilnya dengan menggunakan matlab.

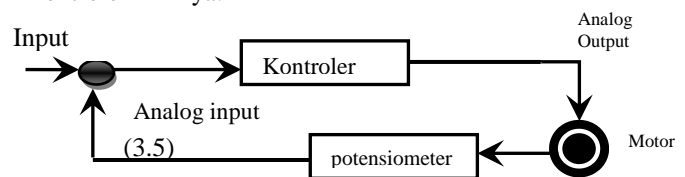


Gambar 3.5 grafik persamaan G(s) dengan menggunakan matlab

Dari gambar diatas dapat kita ketahui waktu $T_p = 0.6$ detik dan %OS 18,2. Hasil respon diatas merupakan pendekatan sistem orde 2 yang ideal. Berbeda dengan percobaan yang dilakukan pada *plant*. Tetapi hasil nya tidak berbeda jauh dengan yang ada pada gambar diatas.

3.3.3 Program kontroler PID

Program kontroler PID yang akan dibuat nantinya bertujuan untuk mengatur besarnya analog *output* dengan masukan dari analog *input*. Berikut blok diagram proses kontroler PIDnya.



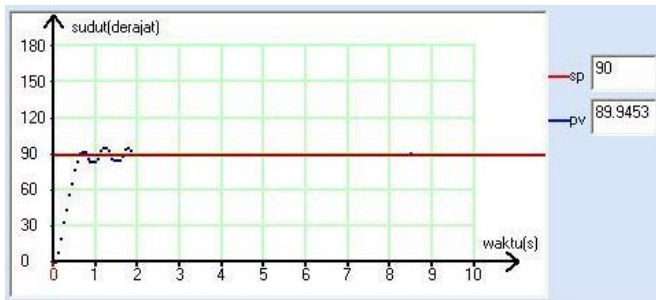
Gambar 3.6 blok diagram system pengaturan posisi

Pada proyek akhir pengaturan posisi motor servo DC ini yang menggunakan kontroler PID, dipergunakan metode untuk mendapat nilai masing-masing parameter K_p , K_i dan K_d nya dengan metode osilasi. Langkah-langkah metode osilasi sebagai berikut :

1. Buat suatu sistem loop tertutup dengan kontroler P dan *plant* di dalamnya.
2. Tambahkan nilai K_p sampai sistem berosilasi berkesinambungan
3. Dapatkan responnya, tentukan nilai K_u dan P_u
4. Tentukan nilai K_p , K_i , dan K_d berdasar tabel 3.1 berikut.

Tabel 3.1 penalaan dengan metode osilasi

Tipe Kontroler	K_p	T_i	T_d
P	$0,5 K_u$	-	-
PI	$0,45 K_u$	$0,5 P_u$	-
PID	$0,6 K_u$	$0,5 P_u$	$0,125 P_u$



Gambar 3.7 respon saat $K_u = 4$

Dari gambar respon diatas dapat diambil data saat $K_u = 4$ dan nilai $P_u = 0.6$ detik. Dengan menggunakan tabel 3.1 penalaan dengan metode osilasi didapatkan hasil nilai dari masing-masing parameter sebagai berikut.

Tabel 3.2 hasil perhitungan dari masing-masing parameter

Tipe Kontroler	K_p	K_i	K_d
P	2	-	-
PI	1.8	4.8	-
PID	2.4	8	0,18

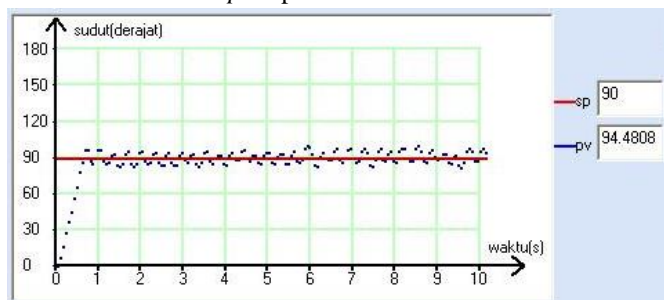
Akan tetapi hasil dari parameter diatas belum tentu bisa bagus responnya. Setelah mendapat masing-masing parameter K_p , K_i , dan K_d masing-masing parameter tersebut masih dituning lagi dengan trial and error sampai respon yang didapat bagus hasilnya.

IV. PENGUJIAN DAN ANALISA

Pada pengujian sistem ini akan dilakukan beberapa pengujian pada respon sistem plant antara lain :

1. Pada saat *close loop* tanpa kontroler
2. Dengan kontroler Proportional
3. Dengan kontroler Proportional-Integral
4. Dengan kontroler PID
5. Dengan kontroler PID yang ditambah beban

1. Pada saat *close loop* tanpa kontroler



Gambar 4.1 respon output saat close loop

Pada gambar diatas merupakan hasil *output* respon sistem dengan mulai pada sudut 0° menuju ke sudut 90° . Pada gambar tersebut respon *output* berosilasi secara terus menerus

dan tidak bisa *steady state* pada *setpoint*. Karena sistem diatas dalam keadaan *close loop* tanpa adanya kontroler apapun.

2. Dengan kontroler proportional

Tabel 4.1 tabel pengujian dengan menggunakan kontroler Proportional

Sudut mulai	target	Pembacaan pada busur derajat	Keterangan
0°	50°	49°	Sistem berosilasi
50°	90°	90°	Sistem berosilasi
90°	120°	120°	Rise time = 0.5 detik, steady state = 0.6 detik
120°	150°	150.5°	Sistem berosilasi tetapi mengalami steady state sekitar 1.6 detik
150°	180°	180°	Sistem berosilasi tetapi mengalami steady state sekitar 2 detik

Pada tabel diatas merupakan tabel pengujian dengan menggunakan kontroler proportional. Dengan menggunakan nilai $K_p = 2$ bisa kita analisa banyak respon sistem yang mengalami osilasi. Secara umum saat hanya menggunakan kontroler P maka respon akan mengalami osilasi. Hanya satu kali percobaan sistem yang tidak mengalami osilasi yaitu saat sudut 90° menuju 120° . Hal ini terjadi putaran motor saat sudut besar dari 90° sampai 180° sedikit lebih berat dibandingkan saat sudut kecil. Dapat kita lihat dari tabel diatas respon sistem saat sudutnya dibawah 90° selalu terjadi osilasi.

3. Dengan kontroler Proportional-Integral

Tabel 4.2 tabel pengujian dengan menggunakan kontroler Proportional-Integral

Sudut mulai	target	Pembacaan pada busur derajat	Keterangan
0°	50°	49°	Sistem berosilasi tetapi mencapai steady state sekitar 5 detik
50°	90°	90°	Sistem berosilasi tetapi mencapai steady state sekitar 2.5 detik
90°	120°	120°	Rise time = 0.4 detik, steady state = 0.5 detik
120°	150°	150.5°	Rise time = 0.8 detik, steady state = 0.9 detik
150°	180°	180°	Rise time = 1.2 detik, steady state = 1.7 detik

Pada tabel diatas merupakan tabel pengujian dengan menggunakan kontroler proportional-integral. Dengan menggunakan nilai $K_p = 0.03$ dan $K_i = 0.01$. Secara umum saat menggunakan kontroler PI respon bisa mencapai *setpoint*. Meskipun dalam tabel diatas ada sistem yang masih berosilasi.

4. Dengan kontroler PID

Tabel 4.3 tabel pengujian dengan menggunakan kontroler PID

Sudut mulai	target	Pembacaan pada busur derajat	Keterangan
0°	50°	49°	Sistem berosilasi tetapi mencapai <i>steady state</i> sekitar 3.4 detik
50°	90°	90°	Sistem berosilasi tetapi mencapai <i>steady state</i> sekitar 4.2 detik
90°	120°	120°	Sistem berosilasi tetapi mencapai <i>steady state</i> sekitar 1.9 detik
120°	150°	150.5°	<i>Rise time</i> = 0.5 detik, <i>steady state</i> = 0.6 detik
150°	180°	180°	<i>Rise time</i> = 0.8 detik, <i>steady state</i> = 0.9 detik

Pada tabel diatas merupakan tabel pengujian dengan menggunakan kontroler proportional-integral. Dengan menggunakan nilai $K_p = 0,09$, $K_i = 0,09$ dan $K_d = 0,03$. Secara umum saat menggunakan kontroler PID respon bisa mencapai *setpoint*. Meskipun dalam tabel diatas ada sistem yang masih berosilasi.

Dari pengujian beberapa kontroler didapatkan penggunaan kontroler PID lebih bagus dibanding dengan yang lain. Meskipun respon ada yang berosilasi namun waktu *steady state* nya lebih cepat dibandingkan dengan kontroler P dan PI.

5. Dengan kontroler PID yang ditambah beban

Tabel 4.4 pengujian dengan pembebanan

Jenis kontroler	Sudut awal-sudut target	1 timbal	2 timbal	3 timbal
P	0°-90°	Respon berosilasi mencapai <i>steady state</i> 4.2 detik	Respon sedikit mengalami <i>overshoot</i> <i>rise time</i> 1.2 detik dan <i>steady state</i> 1.3 detik	<i>Rise time</i> 1.5 detik dan <i>steady state</i> 1.6 detik
PI	0°-90°	respon berosilasi mencapai <i>steady state</i> 9 detik	Respon berosilasi	Respon berosilasi <i>steady state</i> 5.1 detik
PID	0°-90°	Respon berosilasi	Respon berosilasi	<i>Rise time</i> 1.6 detik <i>steady state</i> 1.7 detik

Pada tabel 4.4 diatas merupakan tabel dengan pemberian beban pada masing-masing kontroler. Pada uji pembebanan dengan menggunakan kontroler P saat uji pembebanan dengan 1 timbal, 2 timbal dan 3 timbal terlihat saat dengan 3 timbal responnya lebih bagus. Dengan *rise time* 1,5 detik dan *steady statenya* 1,6 detik. Hal ini disebabkan saat pembebanan 3 timbal busur derajat yang tadinya berjalan cukup cepat terbebani oleh timbal sehingga membuat busur derajat berputar melambat dan mencapai *setpoint* dengan baik.

Pada uji pembebanan dengan menggunakan kontroler PI saat uji pembebanan dengan 1 timbal, 2 timbal dan 3 timbal terlihat saat dengan 3 timbal responnya lebih bagus dibandingkan dengan saat diuji dengan 1 timbal dan 2 timbal. Dengan *steady statenya* 1,6 detik. Hal ini disebabkan saat pembebanan 3 timbal busur derajat yang tadinya berjalan cukup cepat terbebani oleh timbal sehingga membuat busur derajat berputar melambat dan mencapai *setpoint* meskipun masih terlihat adanya osilasi.

Pada uji pembebanan dengan menggunakan kontroler PID saat uji pembebanan dengan 1 timbal, 2 timbal dan 3 timbal terlihat saat dengan 3 timbal responnya lebih bagus dibandingkan dengan saat diuji dengan 1 timbal dan 2 timbal. Sama saat seperti pengujian dengan kontroler P dan PI, pada saat pembebanan 3 timbal responnya bagus. Dengan *Rise time* 1,6 detik *steady state* 1,7 detik. Hal ini disebabkan saat pembebanan 3 timbal busur derajat yang tadinya berjalan cepat terbebani oleh timbal sehingga membuat busur derajat berputar melambat dan mencapai *setpoint* dengan hasil yang bagus.

BAB V PENUTUP

5.1 KESIMPULAN

Setelah melakukan tahap perencanaan dan pengujian sistem, maka dapat diambil kesimpulan sebagai berikut :

1. Sensor posisi potensiometer didapatkan nilai selisih terbesar pada sudut tertentu (sudut 0° dan 180°).
2. Pada plant yang saya buat putaran motor saat berputar searah jarum jam dan berlawanan jarum jam berbeda.
3. Pada plant ini penggunaan kontroler PI lebih bagus dibandingkan dengan yang lainnya.
4. Error pembacaan sudut pada busur derajat berkisar antara 0° sampai 1.5°
5. Penempatan pembebanan berpengaruh pada errornya sudut pembacaan pada busur derajat.

5.2 SARAN

1. Perlunya penggunaan potensiometer yang benar-benar linear dan presisi sebagai sensor posisinya

DAFTAR PUSTAKA

- [1].<http://learncontrol.files.wordpress.com/2009/07/kendali-posisi-motor-dc.pdf>
- [2].<http://elektro-kontrol.blogspot.com/2011/06/realisasi-kontrol-pid-proporsional.html>
- [3].Ogata, Katsuhiko. 1991.Teknik Kontrol Automatik – terjemahan: Ir. Edi Laksono, Jakarta:Erlangga
- [4].Gunterus, Frans.1994. Falsafah Dasar: Sistem Pengendalian Proses, Jakarta: PT. Elex Media Komputindo